# CSc Practical Worksheet 8

## A'S

## Paper 2

## Topic: Flowcharts and Pseudocode

**1** The following pseudocode represents a simple algorithm.

```
DECLARE NumberFound, Remainder, Number : INTEGER DECLARE
    StartNumber, EndNumber, Divisor : INTEGER

INPUT StartNumber
    INPUT EndNumber
    INPUT Divisor
    NumberFound ← 0
    FOR Number ← StartNumber TO EndNumber
       Remainder ← MODULUS(Number, Divisor)
       IF Remainder = 0
      THEN
         OUTPUT Number
         NumberFound      ← NumberFound + 1
       ENDIF
    ENDFOR
OUTPUT "Count: " & NumberFound
```

For the built-in functions list, refer to the **Appendix** on page 14.

**(a)** Complete the following trace table.

| StartNumber | EndNumber | Divisor | NumberFound | Number | Remainder | Output |
|---|---|---|---|---|---|---|
| 11 | 13 | 2 | 0 | **11** | **1** | |
| | | | | **12** | **0** | **12** |
| | | | **1** | | | |
| | | | | **13** | **1** | |
| | | | | | | **Count: 1** |

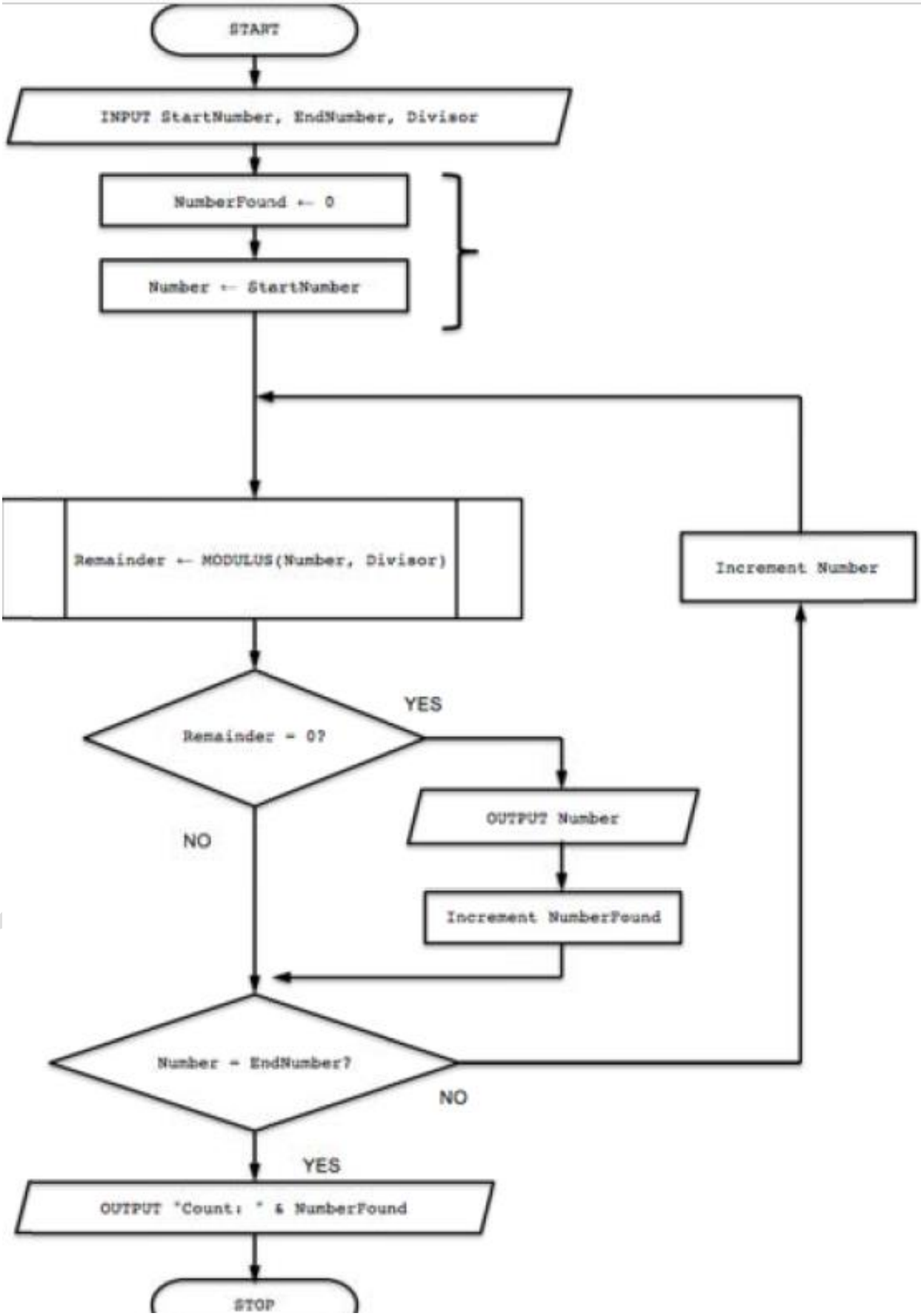
[3]

**(b)** Describe the purpose of this algorithm.

- For a (given) range of values
- Counts the number of times one number (numerator) is an exact divisor of the other
- Outputs each numerator (only)
- Outputs the count

Duke Sir

Draw a program flowchart to represent this algorithm. Variable declarations are **not** required in program flowcharts.




[10]